

MENU

SEARCH

INDEX

DETAIL

1/1



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06083615

(43)Date of publication of application: 25.03.1994

(51)Int.Cl.

G06F 9/30

(21)Application number: 04233959

(71)Applicant:

FUJITSU LTD

(22)Date of filing: 02.09.1992

(72)Inventor:

MARUYAMA TAKUMI

NODA TAKAHITO

KAMISAKA YUJI

NONOMURA KAZUYASU

WATABE TORU

TAKENO TAKUMI

KATO SHINYA

POONSHIYAI

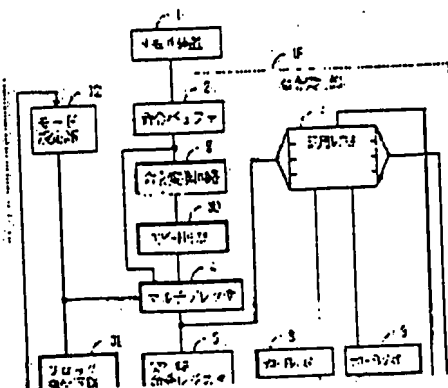
CHIYONSUWANNAPAISAAN

(54) COMPUTER FOR EXECUTING INSTRUCTION SET EMULATION

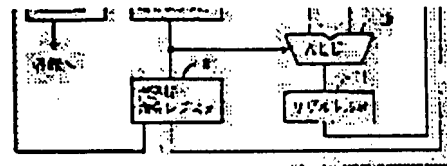
(57)Abstract:

PURPOSE: To provide a computer which can execute the instruction set emulation without causing the down of the machine cycle that performs a basic instruction set.

CONSTITUTION: A computer is provided with a mode setting part 12 which has a basic instruction set and plural different instruction sets and sets an emulation mode and an instruction converting circuit 8 which converts the different instruction sets into a basic instruction set. Thus the computer executes the emulation by converting plural different



instruction sets into a basic instruction set. Furthermore a clock generating circuit 31 is added to the computer to generate the clocks by changing the clock number of an execution stage based on the output of the part 12 and a conversion register 30 which holds the output of the circuit 8. In such constitution, the conversion of instruction are carried out at each independent stage in an instruction set emulation state.



LEGAL STATUS

[Date of request for examination] 25.06.1996

[Date of sending the examiner's decision of rejection] 01.12.1998

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

MENU

SEARCH

INDEX

DETAIL

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-83615

(43)公開日 平成6年(1994)3月25日

(51)IntCl¹

G 0 6 F 9/30

識別記号

3 1 0 E 9189-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 8 頁)

(21)出願番号 特願平4-233959

(22)出願日 平成4年(1992)9月2日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72)発明者 丸山 拓巳

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(72)発明者 野田 敏人

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(72)発明者 神阪 裕士

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74)代理人 弁理士 井桁 貞一

最終頁に続く

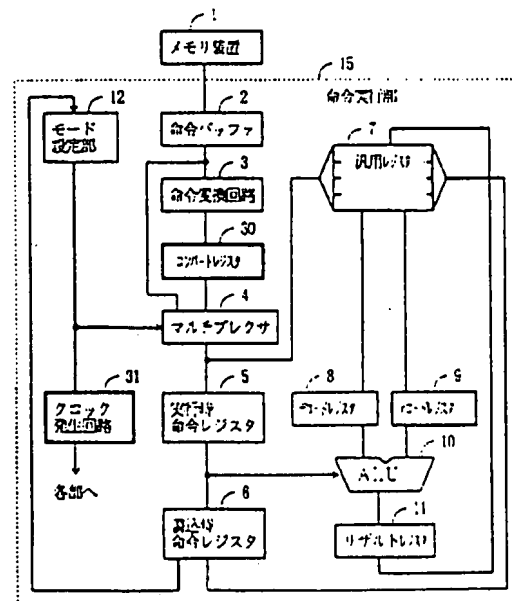
(54)【発明の名称】 命令セットエミュレーションを行う計算機

(57)【要約】

【目的】 命令セットエミュレーションを行う計算機に関し、基本命令セットを実行するマシンサイクルをダウンさせることなく命令セットエミュレートを行うことを目的とする。

【構成】 1つの基本の命令セットと複数の異なる命令セットを持ち、エミュレーションモードを設定するモード設定部と異なる命令セットを当該基本の命令セットに変換する命令変換回路とを有し、複数の異なる命令セットを当該基本の命令セットに変換してエミュレーションを実行する計算機において、前記モード設定部の出力により実行ステージのクロック数を変更して発生するクロック発生回路と該命令変換回路の出力を保持するコンパートレジスタとを設け、命令セットエミュレーション時には、該命令変換を独立した実行ステージで行うように構成する。

本発明の実施例の命令セットエミュレーションの回路構成図



【特許請求の範囲】

【請求項1】 1つの基本の命令セットと複数の異なる命令セットを持ち、エミュレーションモードを設定するモード設定部(12)と異なる命令セットを当該基本の命令セットに変換する命令変換回路(3)とを有し、複数の異なる命令セットを当該基本の命令セットに変換してエミュレーションを実行する計算機において、前記モード設定部(12)の出力により実行ステージのクロック数を変更して発生するクロック発生回路(31)と該命令変換回路(3)の出力を保持するコンバートレジスタ(30)とを設け、命令セットエミュレーション時には、該命令変換を独立した実行ステージで行うことを特徴とする命令セットエミュレーションを行う計算機。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は命令セットエミュレーションを行う計算機に関する。近年のコンピュータシステムの高速化の要求に伴い命令フォーマットの簡素化が求められているが、命令フォーマットの変更はソフトウェアの変更を必要とする。しかし、ソフトウェアの新規開発には多大の作成工数がかかり、かつソフトウェア技術者の不足が叫ばれている今日ではそのような方法は妥当でない。

【0002】従って新しい計算機を開発するには、ソフトウェア互換性を維持しながら命令フォーマットの簡素化、統一を行う必要がある。ここで、ソフトウェアの互換性を保証するためには、複数の命令セットを実行しうる能力を新たに開発する計算機が持つ必要がある。本発明は、この様な命令セットを複数持つ計算機に於ける命令セットエミュレーション手段を提供するものである。

【0003】

【従来の技術】図3は従来の命令セットエミュレーションの回路構成図である。この例においては1命令がフェッチ・デコード・実行・書込の4ステージで実行される。また、本例はパイプライン処理を行う構成であるがエミュレーションにパイプライン処理が必要であることを示すものではない。

【0004】図において、1はメモリ装置であって、計算機のプログラムやデータを記憶するものである。2は命令バッファ(Instruction Buffer)であって、メモリ装置1から読み出した命令を保持するものである。3は命令変換回路であって、命令バッファ2に保持された基本命令セット以外の命令を基本命令セットのいずれかに変換する回路である。4はマルチプレクサであって、命令バッファ2の出力と命令変換回路3の出力のいずれかをモード設定部12の出力により選択して出力する回路である。5は実行時命令レジスタ(Instruction Register)であって、マルチプレクサ4の出力を格納するものであり、実行ステージで使用する命令レジスタである。6は書込時命令レジスタであって、書込ステージで使用する

命令レジスタである。7は汎用レジスタ(General Registers)であり、複数のレジスタファイル素子から構成されるものである。

【0005】8、9はデコードレジスタ(Decode Register)であり、汎用レジスタから読み出したALU計算に必要なレジスタの値を保持するものである。10は演算回路(ALU)であり、演算を行う回路である。11はリザルトレジスタ(Result Register)であって、演算結果を保持するレジスタである。12はモード設定部であって、エミュレーションを行うモードであることを示す回路であって、基本命令でセット及びリセットすることができる。15は命令実行部であって、メモリ装置1からフェッチした命令を解読して実行する部分であり、計算機の中央処理装置(CPU)の要部である。

【0006】本例の計算機では電源が投入されたとき図示されてない初期クリア回路が動作してモード設定部12は基本命令モードに設定される。続いて基本命令モードで記述されたオペレーティングシステムが、図示されていない外部記憶装置から読み込まれて、動作を開始する。命令処理は以下のシーケンスで行われる。

【0007】A. 基本命令セットの場合

この場合にはエミュレーションモードでないのでモード設定は行わない。

① 命令をメモリ装置1から命令バッファ2にフェッチする。

【0008】② 命令バッファ2中のソースレジスタ番号(演算対象が格納されているレジスタの番号)が示すエントリの値を汎用レジスタ7のレジスタファイルから取り出し、デコードレジスタ8,9に格納する。汎用レジスタ7はデュアルポートRAMを使用し同時に2エントリをアクセスできる。同時に命令バッファ2の内容をマルチプレクサ4を通して実行時命令レジスタ5に格納する。

【0009】③ 実行時命令レジスタ5の内容に従った演算をデコードレジスタ8,9に格納した内容に対してALU10に行わせ、結果をリザルトレジスタ11に格納する。同時に実行時命令レジスタ5の内容を書込時命令レジスタ6に格納する。

【0010】④ リザルトレジスタ11の値を、書込時命令レジスタ6中のデスティネーションレジスタ番号(演算結果を格納するレジスタの番号)が示す汎用レジスタ7の該当箇所に格納する。

【0011】図4は従来の命令制御エミュレーションのタイムチャートであり、図のAは上記の基本命令セットの命令実行の動作を示す。上記説明の如く4ステージで1命令の実行が完了する。

【0012】B. 基本命令セット以外の命令セットの場合

基本命令セット以外の命令セットを実行する場合にはエミュレーションモードで動作することを指定するため、

命令セットの一つでモード設定部12をセットする。この場合はオペレーティングシステムのタスク管理機能が、古いアーキテクチャのプログラムを実行することを知り、当該プログラムをメモリ装置1にローディングして、当該プログラムに制御権を渡すと共にモード設定部12をセットする。

【0013】エミュレーションモードの命令処理は以下のシーケンスで行われる。

① 命令をメモリ装置1から命令バッファ2に読み出す。

② 命令バッファ2の内容を命令変換回路3によって基本命令セットに変換し、その結果のマルチプレクサ4の出力が示すソースレジスタ番号のデータを汎用レジスタ7のレジスタファイルから取り出す。

【0014】ここで行われる変換の内容を説明する。従来の命令セットは例えばメモリ領域の節約のために、1バイト長の命令、2バイト長の命令、4バイト長の命令があったが、ICメモリの普及、低価格化のため寧ろ性能向上を課題とするため、命令の長さを4バイトにそろえたような命令セットの変更を行った場合には、その変更は例えばアドレスのディスプレースメントの長さを1バイトから2バイトに変更することであり、上位1バイトにオール0を入れる操作が行われる。

【0015】以下の実行ステージは基本命令セットの場合と同様である。上記の如く制御することによって、基本命令セット以外の命令セットのエミュレーションを実行することができる。また、エミュレーションを終了するときは複数の命令セット中に追加した命令でモード設定部12をリセットすることで基本命令セットに戻る。

【0016】この方法でエミュレーションを実行すると、デコードステージで変換とデコードを実行するため、一つのステージの時間を変換に要する時間だけ延長する必要がある。

【0017】そのため、基本命令セットを実行する時間が延長するという欠点がある。

【0018】

【発明が解決しようとする課題】従来の技術では、命令セットの変換と変換後命令のデコードを同一ステージで行うため、計算機の演算時間が延長する要因となるという問題があった。

【0019】本発明はこのような点にかんがみて、基本命令セットを実行するクロックサイクルをダウンさせることのない命令セットエミュレートを行う手段を提供することを目的とする。

【0020】

【課題を解決するための手段】上記の課題は下記の如くに構成された命令セットエミュレーションを行う計算機によって解決される。

【0021】図1は、本発明の構成図である。1つの基本命令セットと複数の異なる命令セットを持ち、エミ

ュレーションモードを設定するモード設定部12と異なる命令セットを当該基本命令セットに変換する命令変換回路3とを有し、複数の異なる命令セットを当該基本命令セットに変換してエミュレーションを実行する計算機において、前記モード設定部12の出力により実行ステージのクロック数を変更して発生するクロック発生回路31と該命令変換回路3の出力を保持するコンバートレジスタ30とを設け、命令セットエミュレーション時には、該命令変換を独立した実行ステージで行うように構成する。

【0022】

【作用】クロック発生回路31はモード設定部12の出力により実行ステージのクロック数を変更して発生する。エミュレーションを実行するときは変換ステージのために1クロック分余計に発生させる。

【0023】コンバートレジスタ30は命令変換回路3の出力を保持することで変換ステージの結果を保持する。そこで、モード設定部12の出力によりクロック発生回路31は1クロック分余計にクロックを発生させて変換ステージを発生させ、コンバートレジスタ30に変換ステージの結果を保持するようにする。

【0024】従って、変換ステージを独立に設けることができ、ステージの時間を延長させる必要がなくなり、基本命令の実行時間を低下させなくてもよくなる。

【0025】

【実施例】図1に本発明の実施例の命令セットエミュレーションの回路構成図を示す。本実施例においては基本命令セットと基本命令セット以外の命令セットにおいてパイプラインステージ数が異なる。前者はフェッチ・デコード・実行・番込の4ステージ、後者はフェッチ・変換・デコード・実行・番込の5ステージとする。

【0026】図において、30はコンバートレジスタ(Conver Register)であり、エミュレート対象の命令を基本命令セットの命令に変換後の基本命令セットの命令を保持するレジスタである。31はクリア発生回路であって、モード指定部12の出力によりエミュレーションを実行するとき実行ステージのクロック数を変更して変換ステージのために1クロック分余計に発生させる回路である。その他、図3と同一符号の物は同一物である。

【0027】本発明の実施例の回路の動作を説明する。電源投入から動作開始までは従来と同様である。

A. 基本命令セットの場合

基本命令セットの場合は1ステージの時間はデコードステージに変換を行わないためクロックサイクルを延長する必要がない。

【0028】① 命令をメモリ装置1から命令バッファ2にフェッチする。

② 命令バッファ2中のソースレジスタ番号が示す汎用レジスタ7の値をレジスタファイルから取り出し、デコードレジスタ8,9に格納する。同時に命令バッファ2の

内容をマルチプレクサ4を経由して実行時命令レジスタ5に格納する。

【0029】③ 実行時命令レジスタ5の内容に従った演算をALU10で行い、結果をリザルトレジスタ11に格納する。同時に実行時命令レジスタ5の内容を書込時命令レジスタ6に格納する。

【0030】④ リザルトレジスタ11の値を、書込時命令レジスタ6中のデスティネーションレジスタ番号が示す汎用レジスタ7の該当エントリに格納する。このように動作させることによって従来の装置では交換のために延長していた1ステージの時間を延長させることなく基本命令セットの実行ができる。

【0031】図2は本発明の実施例の命令セットエミュレータのタイムチャートである。図のAは基本命令セットの命令実行がフェッチ・デコード・実行・書込の4ステージで完了することを示す。

【0032】B. 基本命令セット以外の命令セットの場合

エミュレーションモードに切り替える動作については従来の動作と同一であるので省略する。

【0033】① 命令をメモリ装置1から命令バッファ2にフェッチする。

② 命令バッファ2の値を命令変換回路3で基本命令セットに変換し、変換結果をコンバートレジスタ30に格納する。

【0034】③ コンバートレジスタ30中のソースレジスタ番号が示す汎用レジスタ7の値をレジスタファイルから取り出し、デコードレジスタ8,9に格納する。同時にコンバートレジスタ30の内容をマルチプレクサ4を経由して実行時命令レジスタ5に格納する。

【0035】以降の動作は従来の動作と同じであり省略

する。図2のBは基本命令セット以外の命令実行がフェッチ・変換・デコード・実行・書込の5ステージで完了することを示す。

【0036】

【発明の効果】本発明によれば、命令セットの交換を独立したステージで行いかつ基本命令セット実行時は当ステージを省くことにより、基本命令セットの性能を落とさない命令セットエミュレートを行う計算機を提供することができる。

10 【図面の簡単な説明】

【図1】 本発明の実施例の命令セットエミュレーションの回路構成図

【図2】 本発明の実施例の命令セットエミュレータのタイムチャート

【図3】 従来の命令セットエミュレーションの回路構成図

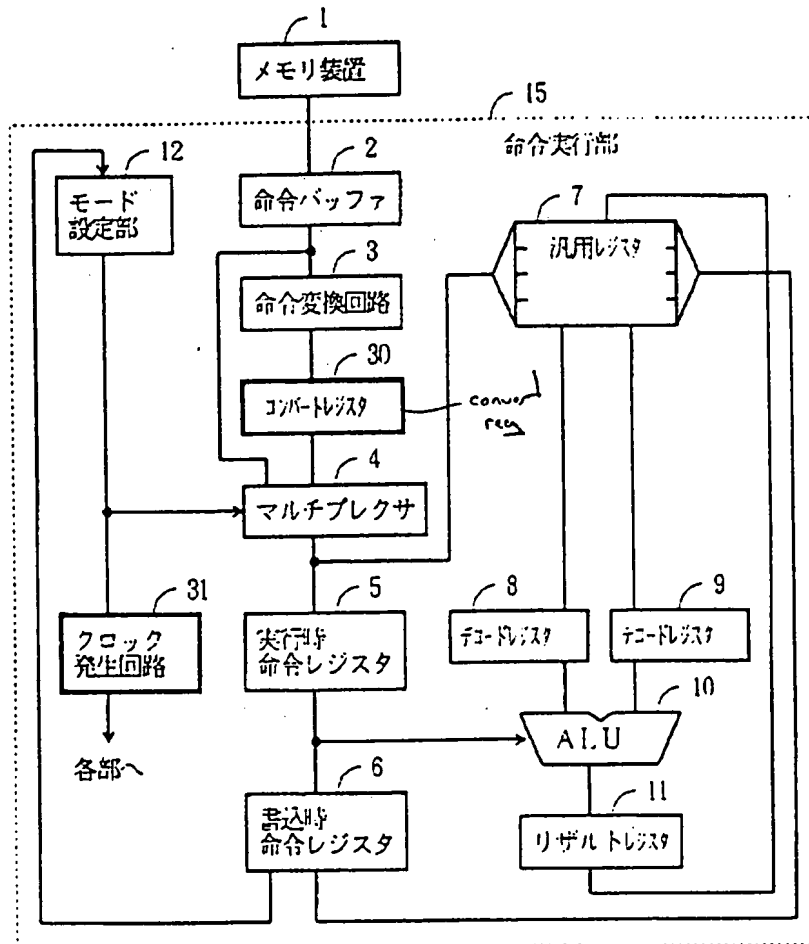
【図4】 従来の命令セットエミュレーションのタイムチャート

【符号の説明】

- | | | | |
|------|-----------|------|-----------|
| 20 1 | メモリ装置 | 2 | 命令バッファ |
| 3 | 命令変換回路 | 4 | マルチプレクサ |
| 5 | 実行時命令レジスタ | 6 | 書込時命令レジスタ |
| 7 | 汎用レジスタ | 8, 9 | デコードレジスタ |
| 10 | ALU | 11 | リザルトレジスタ |
| 12 | モード設定部 | 15 | 命令実行部 |
| 30 | コンバートレジスタ | 31 | クロック発生回路 |

【図1】

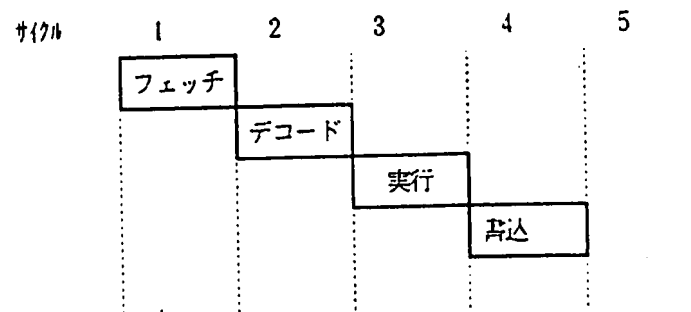
本発明の実施例の命令セットエミュレーションの回路構成図



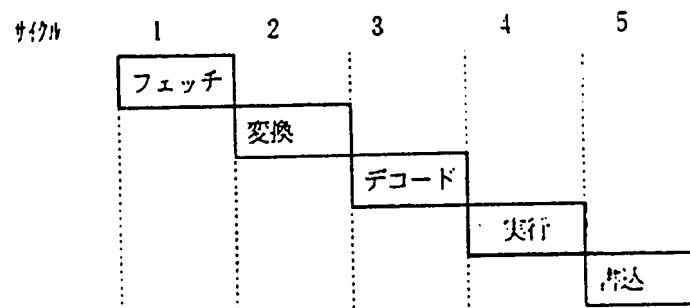
【図2】

本発明の実施例の命令セットエミュレータのタイムチャート

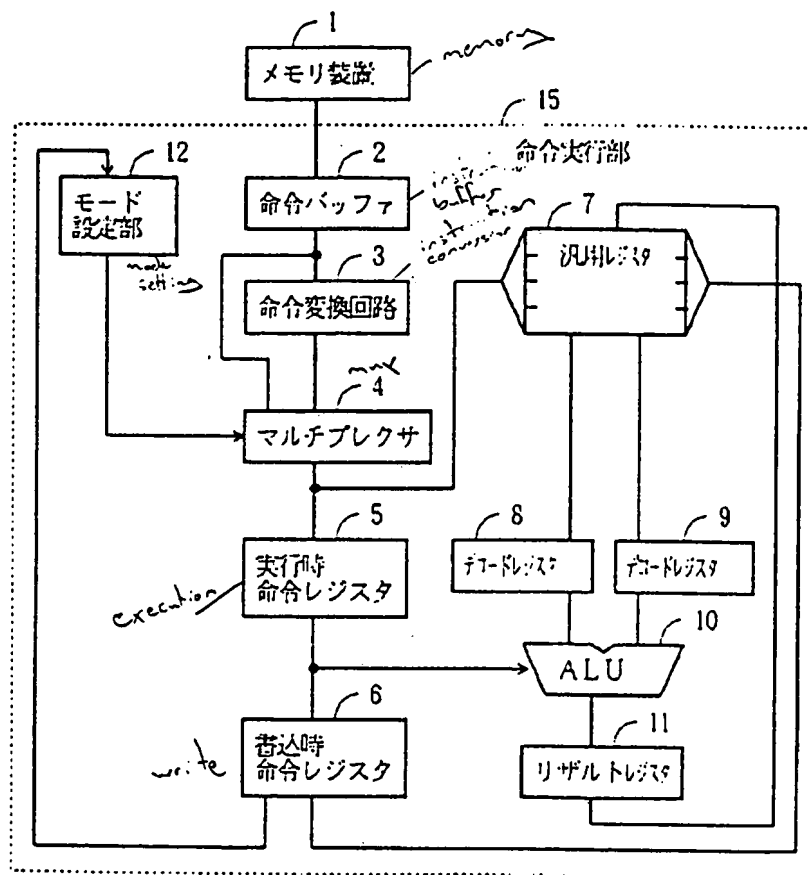
A. 基本命令セットの命令実行



B. 基本命令セット以外の命令実行



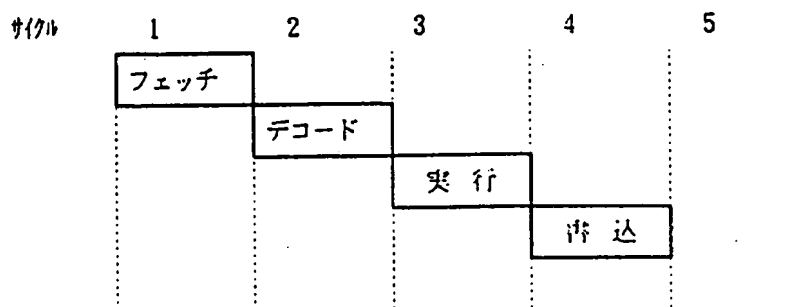
従来の命令セットエミュレーションの回路構成例



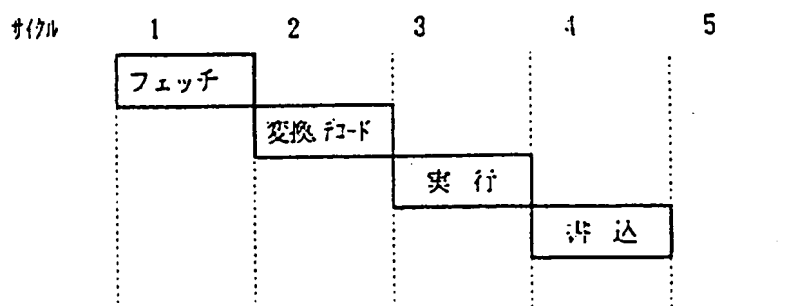
【図4】

従来の命令セットエミュレーションのタイムチャート

A. 基本命令セットの命令実行



B. 基本命令セット以外の命令実行



フロントページの続き

(72)発明者 野々村 一泰

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者 渡部 徹

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者 竹野 巧

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者 加藤 慎哉

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者 ボーンシャイ・チョンスワンナバイサーン

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

An English Translation of JP-A-6-83615

- 1 -

[Scope of Claim for Patent]

- [Claim 1] In a computer having an instruction conversion circuit (3) having one basic instruction set and a plurality of mutually different instruction sets converting an instruction set different from a mode setting portion (12) setting an emulation mode into a basic instruction set, a plurality of mutually different instruction sets are converted into the basic instruction set for executing emulation, characterized in that
- 10 a clock generation circuit (31) generating clock with modifying clock number for execution stage by output of the mode setting portion (12) and convert register (30) holding the output of the instruction converting circuit (3) are provided for performing said instruction conversion in an independent
- 15 execution stage upon instruction set emulation.

[Detailed Description of the Invention]

[0001]

- [Field of Industrial Application] The present invention relates to a computer for executing an instruction set emulation.
- 20 Associating with demand for speeding up of the computer system in the recent years, simplification of instruction format is required. Modification of the instruction format requires modification of the software. However, newly developing software requires huge preparation manpower and is not
- 25 reasonable method for nowadays where lacking of software engineers is a hot topic.

- [0002] Accordingly, for developing new computer, it has been required to simplify and unify instruction format with maintaining compatibility of software. Here, for guarantee
- 30 compatibility of software, it is required that the newly developed computer has capacity for executing a plurality of instruction set. The present invention provides instruction set emulation means in the computer having a plurality of such instruction sets.

35 [0003]

[Prior Art] Fig. 3 is a constructional illustration of a conventional circuit for instruction set emulation. In this example, it is assumed that one instruction is executed through

four stages of fetching, decoding, executing and writing. On the other hand, this example has a construction for performing pipeline process, and shows that pipeline process requires emulation.

5 [0004] In the drawing, 1 denotes a memory device which stores programs of the computer and data. 2 denotes an instruction buffer which holds instruction read out from the memory device 1. 3 denotes an instruction conversion circuit which is a circuit for converting instruction other than the basic
10 instruction set into any of the basic instruction sets. 4 denotes a multiplexer which is a circuit selecting one of an output of the instruction buffer 2 and an output of the instruction conversion circuit 3 depending upon an output of a mode setting portion 12. 5 denotes an instruction register
15 upon execution, which is an instruction register storing the output of the multiplexer to be executing in the execution stage. 6 denotes instruction register for writing which is the instruction register to be used in the writing stage. 7 denotes a general register which is constructed with a plurality of
20 register file elements.

[0005] 8, 9 denote decode registers which hold values of registers necessary for an ALU calculation read out from the general register. 10 denote an arithmetic circuit (ALU) which is a circuit performing arithmetic operation. 11 denotes a result
25 register which is a register holding the result of arithmetic operation. 12 denotes a mode setting portion, which is a circuit representing a mode executing emulation which can be set and reset by the basic instruction. 15 denotes an instruction executing portion, which is a portion decoding instruction
30 fetched from the memory device 1 for execution, which is a major part of the central processing unit (CPU) of the computer.

[0006] In the shown example of the computer, upon turning on of a power source, a not shown initial clear circuit is operated to set the mode setting portion at the basic instruction mode.
35 Subsequently, the operating system described in the basic instruction mode is read from a not shown external storage device to initiate operation. The instruction process is executed through the following sequence.

[0007] A. In case of Basic Instruction Set

In this case, for other than emulation mode, mode setting is not performed.

① The instruction is fetched from the memory device 1 to the instruction buffer 2.

[0008] ② The value of the entry indicative of the source register number (number of the register storing the calculation object) in the instruction buffer 2 is taken out from the register file of the general register 7 to store in the decode registers 8, 9. The general register 7 can access two entries simultaneously using a dual port RAM. At the same time, the content of the instruction buffer 2 is stored in the instruction register 5 for execution through the multiplexer 4.

[0009] ③ Arithmetic operation according to the content of the instruction register 5 for execution is performed by the ALU 10 for the content stored in the decode registers 8, 9. At the same time, the content of the instruction register 5 for execution is stored in the instruction register 6 for writing.

[0010] ④ A value of the result register 11 stores in the corresponding portion of the general register represented by a destination register number (number of register for storing the result of arithmetic operation) in the writing instruction register 6.

[0011] Fig. 4 is a timing chart of the conventional instruction control emulation. A in the drawing represents the operation of instruction execution of the foregoing basic instruction set. As set forth above, execution of one instruction is completed through four stages.

[0012] B. In case of Instruction Set Other Than Basic Instruction Set

Upon execution of the instruction set other than the basic instruction set, in order to designate operation in the emulation mode, the mode setting portion 12 is set with one of the instruction sets. In this case, a task management function of the operating system knows execution of program of old architecture to load the program to the memory device 1 to transfer the control right to the program, and in conjunction therewith the mode setting portion 12 is set.

[0013] Instruction process of the emulation mode is performed in the following sequence.

① Instruction is read out to the instruction buffer 2 from the memory device 1.

5 ② The content of the instruction buffer 2 converts the basic instruction set by the instruction conversion circuit 3 to take out data of the source register number indicative of output of the multiplexer 4 of the result from the register file of the general register 7.

10 [0014] Discussion will be given for the content of conversion performed herein. For example, in order to save the memory region, the conventional instruction sets are one byte length instruction, two bytes length instruction and four bytes length instruction. For taking improvement of performance for
15 spreading and lowering of price of IC memory as task, when modification of the instruction sets for unifying the length of the instruction to four bytes, the modification is to modify the length of the displacement of the address from one byte to two bytes, for example. Then operation to add all 0 for
20 upper one byte.

[0015] The following execution stage is similar to the case of the basic instruction set. By performing control as set forth above, emulation of the instruction set other than the basis instruction set can be executed. On the other hand, when
25 the emulation is terminated, the process is returned to the basic instruction set by resetting the mode setting portion as with the instruction added among a plurality of instruction sets.

[0016] When emulation is executed in this method, conversion
30 and decoding is executed in the decoding stage, a period of one stage is required to extend the period required for conversion.

[0017] Therefore, it encounters a drawback to extend the period for executing the basic instruction set.

35 [0018]

[Problem to be Solved by the Invention] In the prior art, conversion of the instruction set and decoding of the instruction after conversion are performed in the same stage, a problem

is encountered to be a factor for expending the operation period of the computer.

[0019] In view of the problems, an object of the present invention is to provide means for executing instruction set emulation without causing down of the clock cycle.

[0020]

[Means for Solving the Task] The foregoing problem is solved by the computer for executing emulation of the instruction set constructed set forth below.

10 [0021] Fig. 1 is a constructional illustration of the present invention. The invention is constructed that, in a computer having an instruction conversion circuit having one basic instruction set and a plurality of mutually different instruction sets converting an instruction set different from
15 a mode setting portion setting an emulation mode into a basic instruction set, a plurality of mutually different instruction sets are converted into the basic instruction set for executing emulation, a clock generation circuit generating clock with modifying clock number for execution stage by output of the
20 mode setting portion and convert register holding the output of the instruction converting circuit are provided for performing said instruction conversion in an independent execution stage upon instruction set emulation.

[0022]

25 [Operation] The clock generation circuit 31 generates the clock with modifying the clock number of the execution state by the output of the mode setting portion 12. Upon execution of emulation, one extra clock is generated for the conversion stage.

[0023] The convert register 30 holds the result of conversion stage by holding the output of the instruction conversion circuit 3. Therefore, the clock generation circuit 31 generates the conversion stage by generating the one extra clock by the output of the mode setting portion 12 to make the result of conversion stage to be held in the convert register 30.

35 [0024] Accordingly, the conversion stage can be provided independently to make it unnecessary to expend the period of the stage to lower the execution period of the basic instruction.

[0025]

[Embodiment] Fig. 1 is a constructional illustration of the circuit for instruction set emulation of an embodiment of the present invention. In the shown embodiment, number of pipeline stages are different between the basic instruction set and the instruction set other than the basic instruction set. The former is four stages of fetching, decoding, executing and writing, and the latter is five stages of fetching, converting, executing and writing.

5 [0026] In the drawings, 30 denotes a convert register, which is a register for holding instruction of the basic instruction after conversion of the instruction as object for emulation into the instruction of the basis instruction set. 31 denotes a clear generation circuit, which is a circuit for generating one extra clock for the conversion stage by modifying clock number of the execution stage upon execution of emulation by output of the mode designating portion 12. Rest, the components of the same reference numerals as those in Fig. 3 are the same components.

15 [0027] Operation of the circuit of the embodiment of the present invention will be discussed. From on-set of the power source to initiation of operation is the same as those of the prior art.

A. In Case of the Basic Instruction Set

In case of the basic instruction set, a period of one stage is unnecessary to extend the clock cycle since conversion is not performed in the decode stage.

25 [0028] ① The instruction is fetched from the memory device 1 to the instruction buffer 2.

② The value of the general register 7 indicated by the source register number in the instruction buffer 2 is taken out from the register file to store in the decode register 8, 9. At the same time, the content of the instruction buffer 2 is stored in the execution instruction register 5 via the multiplexer 4.

30 [0029] ③ Arithmetic operation according to the content of the execution instruction register 5 by the ALU 10 to store the result in the result register 11. At the same time, the content of the execution instruction register 5 is stored in the writing

instruction register 6.

[0030] ④ The value of the result register 11 is stored in the corresponding entry of the general register 7 represented by the destination register in the writing instruction register 5 6. By operating as set forth above, the basis instruction set can be executed without extending the period for one stage which has been extended for conversion in the conventional device.

[0031] Fig. 2 is a timing chart of the instruction set emulation of the embodiment of the present invention. A in the drawing 10 represents that the basic instruction set is completed through four stages of fetching, decoding, executing and writing.

[0032] B. In Case of Instruction Set Other Than Basic Instruction Set

Switching operation of the emulation mode is omitted for 15 the same as the operation of the prior art.

[0033] ① Instruction is fetched from the memory device 1 to the instruction buffer 2.

② The value of the instruction buffer 2 is converted into the basis instruction set by the instruction conversion circuit 20 3 to store the result of conversion in the convert register 30.

[0034] ③ The value of the general register indicative of the source register number in the convert register is taken out from the register file to store in the decode registers 8, 9/ 25 At the same time, the content of the convert register 30 is stored in the execution instruction register 5 via the multiplexer 4.

[0035] Subsequent operation is the same as that of the prior art and thus is omitted. B of Fig. 2 represents that the 30 instruction execution of other than the basis instruction set is completed through five stages of fetching, converting, decoding, executing and writing.

[0036]

[Effect of the Invention] According to the present invention. 35 by executing the conversion of the instruction set in the independent stage and the conversion stage is omitted upon execution of the basic instruction set, the computer executing emulation of the instruction set without lowering performance

of the basic instruction set can be provided.

[Brief Description of the Drawings]

[Fig. 1] is a constructional illustration of a circuit for instruction set emulation of an embodiment of the present invention;

[Fig. 2] is a timing chart of instruction set emulation of an embodiment of the present invention;

[Fig. 3] is a constructional illustration of the conventional instruction set emulation; and

[Fig. 4] is a timing chart of the conventional instruction set emulation.

[Explanation of Reference Numerals]

- 1 memory device
- 2 instruction buffer
- 15 3 instruction conversion circuit
- 4 multiplexer
- 5 execution instruction register
- 6 writing instruction register
- 7 general register
- 20 8,9 decoder
- 10 ALU
- 11 result register
- 12 mode setting portion
- 15 instruction executing portion
- 25 30 convert register
- 31 clock generating circuit

DRAWINGS

FIG. 1

- 30 1 MEMORY DEVICE
- 12 MODE SETTING PORTION
- 31 CLOCK GENERATING CIRCUIT
- 2 INSTRUCTION BUFFER
- 3 INSTRUCTION CONVERSION CIRCUIT
- 35 30 CONVERT REGISTER
- 4 MULTIPLEXER
- 5 EXECUTION INSTRUCTION REGISTER
- 6 WRITING INSTRUCTION REGISTER

7 GENERAL REGISTER
8 DECODE REGISTER
9 DECODE REGISTER
11 RESULT REGISTER

5

FIG. 2

TIMING CHART OF INSTRUCTION SET EMULATION OF EMBODIMENT OF THE INVENTION

A. INSTRUCTION EXECUTION OF BASIC INSTRUCTION SET

10 1¹ FETCH 2² DECODE 3³ EXECUTE 4⁴ WRITE

B. INSTRUCTION EXECUTION OTHER THAN BASIC INSTRUCTION SET

1¹ FETCH 2² CONVERT 3³ DECODE 4⁴ EXECUTE 5⁵ WRITE

FIG. 3

15 1 MEMORY DEVICE
12 MODE SETTING PORTION
2 INSTRUCTION BUFFER
3 INSTRUCTION CONVERSION CIRCUIT

20 4 MULTIPLEXER
5 EXECUTION INSTRUCTION REGISTER
6 WRITING INSTRUCTION REGISTER
7 GENERAL REGISTER
8 DECODE REGISTER
25 9 DECODE REGISTER
11 RESULT REGISTER

FIG. 4

TIMING CHART OF INSTRUCTION SET EMULATION OF PRIOR ART

30 A. INSTRUCTION EXECUTION OF BASIC INSTRUCTION SET

1¹ FETCH 2² DECODE 3³ EXECUTE 4⁴ WRITE

B. INSTRUCTION EXECUTION OTHER THAN BASIC INSTRUCTION SET

1¹ FETCH 2² CONVERT/DECODE 3³ EXECUTE 4⁴ WRITE

35